REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
	Technical Report	-

4. TITLE AND SUBTITLE
Toward Privacy-preserving Content Access Control for Information Centric Networking

5a. CONTRACT NUMBER
W911NF-11-1-0191

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611102

6. AUTHORS
Bing Li, Zhijie Wang, Dijiang Huang, Yan Zhu

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
Arizona State University
ORSPA
PO Box 876011
Tempe, AZ 85287 -6011

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
56078-CS.5

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
Information Centric Networking (ICN) is a new network architecture that aims to overcome the weakness of existing IP-based end-to-end networking. Instead of knowing the IP address of the communicating party, ICN focuses on the data, i.e. content, transmitted in network. Therefore, how to locate and access the desired content is a crucial issue in ICN. Some existing solutions aim at resolving the content name through a name resolution service, which is similar to the DNS services of Internet. Other solutions are based on route-by-name scheme, which treats

15. SUBJECT TERMS
privacy, naming, information centric networking,access control

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dijiang Huang
UU	UU	UU	UU		19b. TELEPHONE NUMBER 480-965-2776

Report Title

Toward Privacy-preserving Content Access Control for Information Centric Networking

ABSTRACT

Information Centric Networking (ICN) is a new network architecture that aims to overcome the weakness of existing IP-based end-to-end networking. Instead of knowing the IP address of the communicating party, ICN focuses on the data, i.e. content, transmitted in network. Therefore, how to locate and access the desired content is a crucial issue in ICN. Some existing solutions aim at resolving the content name through a name resolution service, which is similar to the DNS services of Internet. Other solutions are based on route-by-name scheme, which treats content names similar to existing routing protocols using IP addresses. Since the content can be cached in various data storage, it is difficult to enforce content access control policies on various content hosting servers. As a result, using Attribute-Based Encryption (ABE) is a flexible approach to enforce the content access policies regardless the security mechanisms provided by different content hosting servers. However, using ABE has a drawback that the enforced content access policies are known to all the ICN users. It is desirable that only legitimated content viewers are able to reveal the content access policies. To this end, a privacy-preserving content access control scheme is presented in this research for ICN. The presented scheme is compatible with existing flat name based ICN architectures.

Toward Privacy-preserving Content Access Control for Information Centric Networking

Bing Li, Zhijie Wang, Dijiang Huang, and Yan Zhu
[1] School of Computing Informatics and Decision Systems Engineering, ASU, USA
[2] School of Computer and Communication Engineering, USTB, China
[1] {bingli5, zhijie.wang, dijiang}@asu.edu, [2]zhuyan@ustb.edu.cn

***Abstract*—Information Centric Networking (ICN) is a new network architecture that aims to overcome the weakness of existing IP-based end-to-end networking. Instead of knowing the IP address of the communicating party, ICN focuses on the data, i.e. content, transmitted in network. Therefore, how to locate and access the desired content is a crucial issue in ICN. Some existing solutions aim at resolving the content name through a name resolution service, which is similar to the DNS services of Internet. Other solutions are based on route-by-name scheme, which treats content names similar to existing routing protocols using IP addresses. Since the content can be cached in various data storage, it is difficult to enforce content access control policies on various content hosting servers. As a result, using Attribute-Based Encryption (ABE) is a flexible approach to enforce the content access policies regardless the security mechanisms provided by different content hosting servers. However, using ABE has a drawback that the enforced content access policies are known to all the ICN users. It is desirable that only legitimated content viewers are able to reveal the content access policies. To this end, a privacy-preserving content access control scheme is presented in this research for ICN. The presented scheme is compatible with existing flat name based ICN architectures.**

***Index Terms*—privacy, naming, information centric networking, access control**

I. Introduction

In the current Internet, if a network entity wants to get the access to some content, it has to locate and connect to the content hosting server based on Internet routing and networking protocols. As a result, the content is associated tightly with the location of the server. The entire network is centered around connecting the content consumers to the content owner. Information such as connection status is important to the success of networking.

Witnessed by the fact that the connection-centered network design is a support for transferring content to the consumers, various ICN architectures [1, 2, 3, 4, 5] are proposed. In ICN architecture, the focus is shifted to connecting the content consumers with the content itself. Thus, instead of identifying the content owner's address, the network changes to identify the authentic content copies. Thus, the consumers do not need to know where the content locates, i.e. the IP address of the content owner. The content name could lead them to a copy of the content. Content owners publishes the content, which could be copied and stored in the network by applying network caches. Network caches are normally storage servers or could be a normal network entity. The purpose of this design is to make sure that the content could be delivered to the consumer with a higher efficiency. For example, it is able to retrieve the nearest (according to some metrics) copy of the content to the consumer. In contrast, in the traditional Internet networking framework, the consumer could only get the content from its owner.

Though the design is efficient in retrieving content using ICN, it brings great challenges to the security issues during content caching and retrieving. One of challenges is that the end-to-end communication security is not easy to support. This is because, in ICN, the consumer cannot predict, from which party it gets the content. Traditional content access control policies cannot be easily enforced by all the content hosting servers when caching the content. Therefore, instead of enforcing the data access control on each content hosting server, a natural approach is to secure the content by enforcing the data access control through cryptographic approaches, i.e., encryption/decryption. Only legitimate users who has proper cryptographic keys can access and then reveal the data content. Since each content is identified by the name, it is easy for any network entity to access the content as long as the name is known. To enforce access control onto the content, several frameworks such as [7] have been proposed. Most of these solutions require additional authorities in network to authenticate each content consumer. These schemes sound but introducing additional network components and complicate the ICN service framework. The reason why it is difficult to establish an access control scheme in ICN systems is that after the content is published, the owner does not have control on the content copies any more. Copies of the content could be scattered around the network. This is different from traditional network where the owner can authenticate the consumer before it provides the content.

To address the data access control problem of ICN, we propose a new content protection scheme to support access control. This approach is inspired by Attribute Based Encryption (ABE) schemes[9, 10, 11]. Instead of incorporating a set of additional components, it only requires one additional trusted third party (TTP) in the network. In addition, it could be seamlessly incorporated into existing flat-name ICN architectures. In our approach, each network entity is assigned with a set of attributes with the help of the TTP according to their real identities and functional attributes. The access control policy for the content is based on combinations of the

attributes in terms of AND and OR operations. This policy is enforced according to the content names instead of the contents. Moreover, the presented solution revises the ABE scheme by hiding the access policies in the encryption. As a result, the privacy-preservation is provided for the content access policies, i.e., only legitimated content viewers can reveal the encryption policies and then decrypt the data content. This feature can greatly improve the privacy protection on ICN data when they are distributed in the public domain. In this way, a user is able to identify its eligibility of the accessed contents through the encrypted names before actually accessing the data content. In summary, the scheme we proposed in this paper achieves the following features:

- It preserves the confidentiality of the access policy of contents. Ineligible consumers cannot derive the data access policies even if they collude together;
- It supports any combination of attributes under AND and OR operations in the access control policies, which make it very flexible to construct a data access policies based on known attributes. As a result, even an eligible consumer may not know the full data access policies after a successful decryption due to the use of OR gates in the encryption policy tree;
- It significantly reduces the computation and communication overhead for a potential consumer to determine whether it is eligible to the access the content;

The remainder of this paper is organized as follows. Section II goes through the related work on ICN and its security. Section III presents the system models and preliminaries. Detailed description of our scheme is provided in Section IV, and its performance and security analysis is given in Section V. We conclude this paper in Section VI.

II. Related Work

In this paper, we will propose an ABE-based scheme to enforce a secure access control mechanism in ICN systems. Before going into details of our approach, we will introduce research results on ICN and ABE respectively.

A. ICN Solutions

Several network architectures have been built in the past years. These approaches are different from each other in several aspects though the main idea is centered around information process and management. Among them, CBCB [1] runs on the application layer. It uses publish/subscribe scheme to publish contents. Each consumer broadcasts its interest in the form of attribute combinations. These interests are propogated through the network. At each router, the interests associated with an interface are updated in the form of predicates. Then when a content is transfered through the network, the content is compared with the predicates on every interface to determine through which interfaces to forward the content.

DONA [2] is an ICN project that is deployed above IP layer. It aims to replace the name resolution system in network. The name of a content is in the form of P:L, where P represents the hash of the owner's public key, L is a unique label the owner assigns to the content. The owner registers the content into the name resolution system when it is ready to publish. The consumers use the name resolution system to find the nearest copy of the content. The system will return with the content copy or the IP address of the content location. NetInf [4] uses a similar naming scheme as DONA. But instead of using the owner's public key to generate the digest, it uses a pair of public/private keys for each content. It also uses multi-level Distributed Hash Table (DHT) for name resolution. A content owner needs to register its content in all the three levels and content lookup is carried out from the lowest level upwards. If it is not successful, then an individual resolution system will be used. PURSUIT [5] also uses a similar naming scheme as DONA. But it has a much different structure for retrieving the content location which involves topology information and load balance. Besides, it uses Bloom filter for source oriented routing to forward the content to its consumer.

Unlike the above solutions, NDN [3] uses human-friendly names instead of flat names. A name in NDN consists of multiple components, each of which is a human-readable string. It also contains a digest of the content. This solution uses the name to execute a routing process that is similar to the current IP-based routing. Tables similar to route tables maintain the prefix of names and the corresponding interfaces or data. In this way, a response to a content request could be the content itself. Also, this solution aims to provide a replacement to IP instead of being a layer above IP, which is different from above approaches.

All these ICN methods focus on the efficiency and security aspects of the network while access control to the content and content privacy are not well studied. In [7], an independent access control system is introduced to support the need in ICN. This system connects to the ICN structure through a component called the Relaying Party (RP). An additional component called Access Control Provider (ACP) is in charge of helping content owners create access policies and enforcing the policies to consumers' credentials. This system incorporates access control into ICN systems but requires more network interactions for a consumer to get the content. For protecting content privacy purposes, [12] proposes a design in which each file is divided into blocks. Two or more blocks are mixed to form a chunk. A block from the file is mixed with blocks from "cover" content using randomizing transformations and the results are published to the network so that the adversary could not retrieve the original file easily. To recover the file, an authentic consumer needs to get more information related to the file from a secure channel. With such information, the consumer requrests related chunks from the network. But the requirement of a secure channel is not quite realistic in many application scenarios.

B. ABE Schemes

ABE schemes are originated from Identity-Based Encryption (IBE) which aims to use the user's id as the public key for asymmetric encryptions. After that, an ABE scheme

named Ciphertext-Policy ABE (CP-ABE) [9] is introduced by J. Bethencourt *et al*. This scheme assigns each user with a set of attributes according to their real life roles and identities. There is one private key corresponding to each attribute. A policy specifying under what condition the ciphertext could be successfully decrypted is constructed by the encryptor. This policy is attached with ciphertext in plaintext. Users who do not possess a satisfactory combination of attributes are not able to decrypt the ciphertext. This scheme enables providing access control to individual messages. A message sender (or a content owner in ICN context) is able to specify the required attribute combinations without having to know the receivers' keys. In addition to this feature, this scheme can defend against colluding attackers.

The reason why CP-ABE is not suitable for ICN usage is that the policy is transmitted in clear text. In this way, any network user who has access to the ciphertext during transmission can access details of the policy. Attackers can deduce the sensitivity of the message as well as inferring the role of those who are involved in the message transmission. For example, a message encrypted with the policy $\{Dean\}\, AND\, \{UniversityPresident\}$ is definitely more important than one with policy $\{Faculty\}\, AND\, \{Student\}$. Thus, attackers can identify those high-value users and concentrate on attack these targets.

What needs to change to CP-ABE is to hide the policy into the ciphertext. For this purpose, several works[10, 13] have made pretty good progresses. An attacker cannot get any information about the policy even if it actually executes the decryption process. But these solutions sacrifice efficiency to security in that any party that tries to decrypt the ciphertext will have to go through the entire decryption process which involves a heavy computation overhead.

To make those unsatisfactory users realize their ineligibility as soon as possible to save computation resources, D. Huang *et al*. proposed a scheme[14] to expose the policy attributes step by step. Only one attribute is exposed to the decryptor at one step. In this way, the decrypter is able to stop the decryption process as soon as it fails at one step. But the price for this ability is that one additional attribute, which is the one that fails the decrypter, is exposed. Besides, this approach does not support OR-gates which limits the flexibility of the policy.

III. Models and Preliminaries

In this section, we present a basic ICN framework model and the corresponding security model.

A. ICN framework model

The content in an ICN system consists of at least two parts: the data to be transferred and some meta-data. The data part of the content can be any file, like a text or a picture, or a chunk of a file. The meta-data part contains authenticity and integrity related information.

In a typical ICN system, there are three main roles of network entities: content owner, content consumer and content cache. A content owner may not be the one who creates the content but it fully possesses the ownership of the content. A consumer is a network entity that requests the content. It needs to get the content from the network with the help of the ICN infrastructure. A cache is an entity that is willing to hold a copy of the content for a period of time in its own local storage for some reasons so that whenever a request for the content arrives, it responds with a copy of the content to the consumer. All these three network roles are exchangeable for individual network entities. That is to say, an entity could be the owner, a cache and a consumer for different contents at the same time (Figure 1).

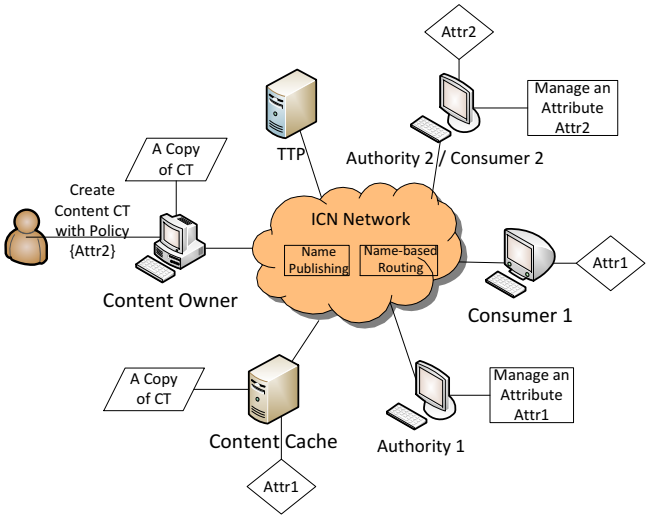

Figure 1: Basic ICN System Model.

The ICN system consists at least two components: a Name Publishing (NP) system and a Name-based Routing (NR) system. The NP is in charge of publishing the content names. The NR is able to retrieve the content based on its network name. Details on how these two systems are realized will not be illustrated in this paper since it is not the focus of the work. Interested reader can refer to [2], [3], and [1] for more information. In addition to these basic parts, our scheme includes a TTP which is trusted to the entire network. The TTP is in charge of setting up ABE-related global parameters for the network. It also helps assigning attributes to individual entities.

An attribute could be any label that is used to identify a person or an entity. In the proposed scheme, every network entity is associated with a unique identifier (UID) and a set of attributes. Here, UID itself can be treated as a special attribute. Attributes (other than UIDs) can be defined and managed by any entity in network. But the definition and management process on an attribute should be carried out by the same entity. This entity is denoted as the authority of the attribute.

Before any entity creates a content, the TTP needs to set up global parameters for the entire ICN system. After that, any entity in the network can create attributes and assign them to

anyone interested in them. Detailed process on how attributes are distributed is out of the scope of this work. Interested reader can refer to allocation problem solutions such as [15]. At this phrase, entities are good to create contents.

When an entity needs to publish a file, as the content owner, it needs to set up an access policy for its content before publishing it. The policy is represented as a combination of related attributes with AND and OR gates. For example, if a content owner wants to create a file that should be accessible only to people working at the HR and the $R\&D$ departments of a company A, then the policy could be $\{A\}\, AND\, \{\{HR\}\, OR\, \{R\&D\}\}$. In this way, the owner does not need to know explicitly who should access the content. All it needs to is to identify the attributes and the combination so that as long as a consumer satisfies the policy, it is able to access the content. Any entity who does not satisfy the policy will not be able to access the file in this content.

After that, the owner generates a random symmetric key and uses this key to encrypt the file to be published. The encryption result is set as the data part of the content. Then the owner creates a name for the content. It uses our scheme to encrypt the random key with the policy it has already specified. The result of this process is used as the real name of the content. Here we need to emphasize that the real name generated using our scheme hides the content access policies so that no one can get the entire policy from the name. The network name, which is used for the ICN system to retrieve the content, is the hash value of the real name. The owner then publishes the real name and the network name of the content into the ICN system. This process is depicted in Figure 2.

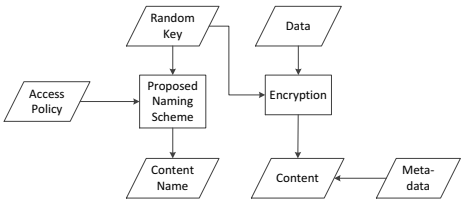

Figure 2: Creating a Content

A consumer who needs this file can get the real name of the corresponding content through the NP system. Before it uses the NR system to get the content, it uses its attributes to decrypt the real name. If its attributes satisfy the hidden policy in the real name, then it can get the random symmetric key protected in the name. Also, it generates the network name and uses it to get the content through the NP system. The data of the content then can be decoded using the random key to get the original file. If a consumer cannot successfully decrypt the real name of the content, then it means the consumer is not allowed to access the original file. Thus, even if it downloads the content using the network name, it still does not have the random key to decode the data.

In the example of Figure 1, there are two attributes Attr1 and Attr2 managed by Authority 1 and Authority 2 respectively. The content owner creates a policy $\{Attr2\}$ to the content published. The content is published into the network and two consumers are willing to get the content. When Consumer 1 gets the real name of the content, it discovers that its attribute Attr1 could not decode the name. Thus, Consumer 1 knows that the content is not intended for it. When Consumer 2 gets the real name, it can successfully decrypt the name. Thus, it generates the network name of the content and using the Name-based Routing system to download the content and uses the random key it gets from the real name to decrypt the data part of the content. Through this figure, we can see that Consumer 2 also acts as the authority of Attr2. A network entity can be a Content Owner, a Content Consumer, a Content Cache and an authority at the same time.

B. Attack model

In order to guarantee the integrity of content, a digital digest signed by the owner is included in the content meta-data. Since data integrity is not the focus of this paper, we will not provide detailed information on this issue.

In the following of this paper, we assume that the attackers have two goals to achieve in compromising the access control scheme: (1) acquiring unauthorized privilege to the data; (2) retrieving constitutional information of access policies so as to gain more information about the content, the content owner and the consumers. The information includes but is not limit to the identity of the owner or consumers, the sensitivity of the content and the potential value of data in the content. For the first goal, the attackers will have to break the confidentiality mechanism of the protected data. Possible methods include exploiting vulnerabilities within the protection functionality of the content. For the second goal, which could be treated secondary to the first one, attackers will have to analyze the ABE-based scheme we propose in this paper so as to identify possible ways to reveal the policy. In order to illustrate our scheme step by step, we firstly introduce basics about CP-ABE which is the origin of our proposed scheme.

C. Preliminaries of CP-ABE

The foundation of CP-ABE is bilinear pairing computation. Let's assume there are two groups: an additive group G_0 and a multiplicative group G_1. They share a same large prime order p. Discrete Logarithm Problem is difficult in both of them. We define a bilinear map $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$. This map has three properties:

- Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$, for any $P, Q \in \mathbb{G}_0$ and $a, b \in \mathbb{Z}_p$;
- Nondegeneracy: $e(g, g) \neq 1$, where g is the generator of $\mathbb{G}_0$;
- Efficiency: Computing the pairing can be efficiently achieved.

In CP-ABE, there are three types of keys: master key, public key and private key. A TTP is required to generate a set of public parameters and securely store the master key. The TTP will not be involved in the network communication. It can be offline all the time. The scheme of CP-ABE consists of four

Table I: Notations

Terms	Meaning
$\mathbb{G}_0$	a bilinear group with a prime order p
$\mathbb{G}_1$	a multiplicative group with the same prime order p
$e(\cdot)$	a bilinear map $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$
$ROOT$	a global constant value $ROOT \in \mathbb{G}_1$ as identification of the secret message protected with the policy
$Enc_k(\cdot)$ $Dec_k(\cdot)$	a symmetric encryption algorithm $Enc_k(\cdot)$ and the corresponding decryption algorithm $Dec_k(\cdot)$ in $\mathbb{G}_1$
encryption sequence	the sequence of attributes in a conjunctive clause in encryption
decryption sequence	the sequence of attributes in a conjunctive clause in decryption
A_i or A_n	an attribute, A_i is used for denoting an individual att--ribute, A_n denotes the n-th attribute in a sequence
A_{Pub}	a public attribute shared among all the network nodes, the corresponding values stored at each node are (I_{Pub}, T_{Pub}), $I_{Pub} \in \mathbb{Z}_p, T_{Pub} \in \mathbb{G}_0$

basic algorithms: **Setup**, **Encrypt**, **KeyGen** and **Decrypt**. In **Setup**, the TTP chooses two random exponents $\alpha, \beta \in Z_p$. A public key is formatted as $< G_0, g, h, f, e(g,g)^{\alpha} >$ while the master key is (β, g^{α}). Here $h = g^{\beta}$, $f = g^{\frac{1}{\beta}}$. The public key is published by the TTP before deployment. When a party wants to encrypt a message M, it runs the **Encrypt** algorithm. The inputs of this algorithm are the public key, the message M and a policy tree T. The output is a ciphertext. The **KeyGen** algorithm is used to generate private keys based on its inputs: the master key and a set of attributes. For each network node, the TTP runs the **KeyGen** algorithm once to generate a private key according to attributes assigned to that node. When a node receives the ciphertext, it runs the **Decrypt** algorithm to get the encrypted data. This algorithm takes the ciphertext and the node's private keys as inputs.

IV. ABE-BASED ICN NAMING SCHEME

In this section, we illustrate our ABE-based naming scheme for ICN network. This scheme is based on previous ABE algorithms [9][14]. Since it is tightly related to attributes, random symmetric keys and attribute keys, we will illustrate the management of these factors as well. Before introducing details of our scheme, we provide a summary of notations in Table I.

A. ABE-based Naming Scheme

!!!!!!

Attributes of an entity can be any value in strings. In CP-ABE, these values are converted into mathematical values with hash functions. In our scheme, each attribute string A_i corresponds to a triplet (T_i, I_i, k_i). The map from a string to such a triplet is not defined by hash functions but determined by the authority of A_i. An access policy can be expressed in Disjunctive Normal Form (DNF) of attributes. In each conjunctive clause of the DNF, the sequence of attributes is enforced by the encryptor. The sequence of encrypting a conjunctive clause is opposite to the sequence of decryption. We name the sequence of encrypting a clause as encryption sequence and the opposite sequence as decryption sequence. We define a public attribute A_{Pub} in our scheme. Unlike other attributes, A_{Pub} is associated with an ordered pair (T_{Pub}, I_{Pub}). For each conjunctive clause, the encryptor adds A_{Pub} at the end of the encryption sequence. Also, the encryptor is required to simplify the DNF so as to reduce the size of attribute policy.

In this scheme, a **GlobalSetup** algorithm is run by a TTP to generate global parameters for the system. For each node joining in the network, the TTP runs **NodeJoin** algorithm once to generate a unique secret for the node. The input of **NodeJoin** is the node's UID while the outputs are $\{D_{UID}$, $X_{Pub,UID}$, Y_{Pub}, $Z_{Pub,UID}\}$. For each attribute, the authority in charge runs an **AuthoritySetup** algorithm to generate secrets associated with that attribute. Besides, our scheme includes other four basic algorithms: **KeyGen**, **Encrypt**, **Decrypt** and **Hash**. The **Encrypt** algorithm will generate results in three different algeobraic structures. The **Hash** algorithm is used to convert the results of **Encrypt** in each algeobraic structure into one element so that the final result is a triplet with each element coming from one algeobraic structure. Since this requirement can be fulfilled by any algeobraic operation in the corresponding structure, we will not provide details of this algorithm.

The **GlobalSetup** algorithm and **NodeJoin** algorithm are defined as in **Algorithm 1** and **Algorithm 2**.

Algorithm 1 GlobalSetup

1: Choose a bilinear group $\mathbb{G}_0$ with a prime order p. p is large enough. g is the generator of $\mathbb{G}_0$;
2: Choose two random values $\alpha, \beta \in \mathbb{Z}_p$;
3: Publicly define a global constant value $ROOT \in \mathbb{G}_1$ as identification of the secret message;
4: Publicly choose a symmetric encryption algorithm $Enc_k(\cdot)$ and the corresponding decryption algorithm $Dec_k(\cdot)$ in $\mathbb{G}_1$;
5: Define and publish a public attribute shared among the network nodes, $(S_{Pub}, T_{Pub}), S_{Pub} \in \mathbb{Z}_p, T_{Pub} \in \mathbb{G}_0$;
6: The global parameters are $\{\mathbb{G}_0, g, g^{\beta}, e(g,g)^{\alpha}, Enc_k(\cdot), Dec_k(\cdot), (S_{Pub}, T_{Pub}), ROOT\}$, global secrets are $\{\beta, g^{\alpha}\}$.

Each individual authority that manages an attribute A_i will have to run **Algorithm 3** to set up attribute secrets.

The **KeyGen** algorithm generates the private keys corresponding to each attribute for each node holding this attribute. It is a cooperative algorithm between an authority and the TTP which is defined in **Algorithm 4**.

The **Encrypt** algorithm works like this: following the encryption sequence of each conjunctive clause, denote each attribute from I_1 to I_m, m is the number of attributes in the clause. Choose a random value $s \in Z_p$ and set $I_0 = s$. Given such a clause, the encryption process on message S_k

Algorithm 2 NodeJoin

1: For each node with UID joining in the network, generate a random number $r_{UID} \in \mathbb{Z}_p$ and store it securely;
2: Calculate and assign $D_{UID} = g^{(\alpha + r_{UID})/\beta}$ to the node;
3: Calculate and assign to the node:

$$X_{Pub,UID} = g^{r_{UID}} T_{Pub}^{r_{Pub}},$$

$$Y_{Pub} = g^{r_{Pub}},$$

$$Z_{Pub,UID} = e(g,g)^{r_{UID} I_{Pub}}.$$

where $r_{Pub} \in \mathbb{Z}_p$ is a random number for each node;
4: Assign to the node $\{D_{UID}, X_{Pub,UID}, Y_{Pub}, Z_{Pub,UID}\}$.

Algorithm 3 AuthoritySetup

1: For each attribute A_i, choose two random numbers $I_i, k_i \in \mathbb{Z}_p$;
2: For each attribute A_i, choose one random value $T_i \in \mathbb{G}_0$.

goes as shown in **Algorithm 5**. A complete encryption process includes such a process for every clause but the overlapping parts of clauses. For example, given a policy {A **AND** B **AND** C} or {A **AND** B **AND** D}, A, B, C, D are four attributes, the simplified form is {A **AND** B} **AND** {C **OR** D}. The encryptor can encrypt {A **AND** B **AND** C} first and then use the results for {A **AND** B} to finish {A **AND** B **AND** D} = {A **AND** B} **AND** {D}.

The **Decrypt** algorithm works in the decryption sequence. Note that the first attribute in decryption sequence is always A_{Pub}. The decrypter follows **Algorithm 6** to conduct decryption.

When **Decrypt** algorithm succeeds, S_k is the group session key embedded in C.

B. Apply ABE-based Naming Scheme in ICN

With the above proposed ABE-based Naming scheme, we can achieve the following abilities:

- Specifying the access control policy without knowing the consumers' keys;

Algorithm 4 KeyGen

1: For each attribute A_i assigned for node with UID, the authority passes UID, I_i and T_i to TTP;
2: TTP computes and sends back to the authority:

$$X_{i,UID} = g^{r_{UID}} T_i^{r_i},$$

$$Y_i = g^{r_i},$$

$$Z_{i,UID} = e(g,g)^{r_{UID} I_i}.$$

where $r_i \in \mathbb{Z}_p$ is a random number;
3: The authority assigns $X_{i,UID}$, Y_i and $Z_{i,UID}$ to the node together with T_i, I_i and k_i.

Algorithm 5 Encrypt

1: Calculate $C = S_k e(g,g)^{\alpha s}$, $C' = g^{\beta s}$ and $C'' = Enc_{S_k}(ROOT)$;
2: Start from the beginning of the clause in encryption sequence;
3: For each attribute A_n, **if** a triplet $(C_{1,n}, C_{2,n}, C_{3,n})$ has already been calculated, move to the next attribute A_{n+1} and restart step 3 with A_{n+1}; **else**, **goto** step 4;
4: Choose a random number $t_n \in \mathbb{Z}_p$;
5: Calculate:

$$C_{1,n} = g^{(I_{n-1} - I_n) t_n},$$

$$C_{2,n} = T_n^{(I_{n-1} - I_n) t_n},$$

$$C_{3,n} = (k_n t_n)^{-1}.$$

$1 \le n \le m$;
6: Calculate $C_{1,m+1} = g^{(I_m - I_{Pub})}$, $C_{2,m+1} = T_{Pub}^{(I_m - S_{Pub})}$.

Algorithm 6 Decrypt

1: Start from the public attribute A_{Pub};
2: For each attribute A_n that the decrypter possesses, compute:

$$\frac{Z_{n,UID_{dec}} \cdot e(X_{n,UID_{dec}}, (C_{1,n})^{k_n C_{3,n}})}{e(Y_n, (C_{2,n})^{k_n C_{3,n}})}$$

$$= e(g,g)^{r_{UID_{dec}} (I_{n-1})};$$

3: **If** $e(g,g)^{r_{UID_{dec}} (I_{n-1})}$ is one of the decrypter's private keys, then go to step 2 with attribute A_{n-1}; **else** go to step 4;
4: Calculate

$$S_k = C / (e(C', D) / e(g,g)^{r_{UID_{dec}} (I_{n-1})}.$$

if $Dec_{S_k}(C'') == ROOT$, **Success**; **else Failure**.

- Full preservation of the policy confidentiality from leaking to adversaries;
- Step-by-step attribute exposure for consumers to determine their eligibility efficiently in computation;
- Flexible attribute management.

C. Attribute Key Update

!!!!!!

In addition to the basic ICN related functions, it is necessary to provide a key update function for attribute keys. The reason is that when a new entity joins in the network after the initial setup, it may be desirable to make sure previous contents are unavailable. Also, it may be true that a certain entity needs to be deprived of an attribute for reasons like dishonest behaviors. In such situations, a key update algorithm is needed for the attribute keys. This algorithm is given in **Algorithm 7**.

Algorithm 7 KeyUpdate

1: For attribute A_i, choose two random values $I_i', \Delta k_i \in \mathbb{Z}_p$;
2: Encrypt I_i' and Δk_i for each intended node UID that has attribute A_i using the node's UID as the policy;
3: Each node updates its keys as $Z'_{i,UID} = (Z_{i,UID})^{I_i'/I_i}$, $k_i' = k_i + \Delta k_i$.

V. Analysis and Evaluation

In this section, the ABE-based naming scheme is evaluated from performance and security aspects. For performance, we analyze its computation consumption and its communication (and storage) overhead. The computation consumtion analysis is carried out by comparing the proposed scheme with existing ABE schemes. The communication comparison is carried out on both the content name and the content itself respectively since they both are transferred in the network. For security issues, we prove the security strength of our ABE-based naming scheme according to the attack model in Section III-B.

A. Performance Analysis

From performance perspective, we are more concerned with the time consumption for a consumer to decode the content's real name. Therefore, we will calculate the time it takes for the decryption process. We treat our algorithm as an encryption algorithm when testing the time consumption. Thus, a comparison on the computation overhead of the proposed scheme with CP-ABE [9], CN scheme [16], NYO scheme (the 2nd construction in [13]) , YRL scheme [10] and GIE scheme [14] is carried out. The idea is to compare the number of time-consuming operations needed in each scheme.

We use a Dell D630 laptop (Intel Core2 Duo T8100 processor 2.10GHz, 1GB memory) with Ubuntu 10.04 for experiment. A Type A pairing with the help of PBC Library [17] is set up. We test every operation for fifty times and choose the average value as basics for our comparison. Results of our experiment (Table II) show that pairing operation takes longer than any other operations.Therefore, our comparasion metric is set to be the number of pairing operations in decryption process.

Table II: Different operations' time-consumption (in milliseconds)

Operation	Pairing	Exponentiation	Multiplication	Inversion
Time	4.574	0.088	0.016	0.038

Following the above-mentioned idea, we use N_{attr} to denote the number of attributes a consumer has. We assume that the total number of attributes defined in the network is N_{all}. Since the policy is publicly known in CP-ABE and CN, decrypters are able to decide what attributes to use in decryption. Therefore, for those who satisfy the policy, the time costed for decryption is proportional to the number of attributes involved, which is denoted as N_{invo}, $N_{invo} \leqslant N_{attr}$. It is obcious that for the unauthentic decrypters, it takes 0 in time since the decrypter would halt the decryption. An unauthentic decrypter in GIE and our scheme is not able to proceed with the decryption process if it cannot meet the next attribute. In this situation, we use N_{part} to denote the number of attributes that the consumer has already decrypted, where $N_{part} \leqslant N_{invo}$. Since OR-gate is not widely supported by all the ABE-schemes we mentioned before, we test the performance with policys consisting attributes and AND-gates only. The result of our test is shown in Table III. We need to point out here that in real world, N_{all} is far larger than N_{attr}. Therefore, CN scheme has the largest cost. Among all the anonymity schemes, GIE and our scheme cost less than NYO and YRL. As a matter of fact, the cost of our scheme is around 2 thirds of that of GIE.

Table III: Comparison of computation cost in decryption

Scheme	Anonymity Supp
CP-ABE	No
CN **CN from $N_{all}+1$ to $N_{path}+1$, need to redraw the figures!!!!!!**	No
NYO	Yes
YRL	Yes
GIE	Yes
Proposed	Yes

The relationship between time consumption and different values of N_{all}, N_{attr} and N_{invo} is illustrated in Figures 3 to 4. We do not provide the relationship with N_{part} because the trend is very close to that with N_{invo}. All these figures are generated by changing one value among N_{all}, N_{attr} and N_{invo} while keeping the other values constant. From these figures, it is clear that when N_{all} or N_{attr} changes, the performance of our proposed scheme does not get influenced. The performance under these two scenarios is the same as that of CP-ABE which are the lowest two schemes in time consumption. This is also applicable to N_{invo} when N_{invo} is less than a certain value, 8 in this specific setting. When N_{invo} gets greater than the threshold, CN scheme becomes the most efficient one. This is because CN scheme uses all the attributes a decrypter has for decryption. The fact that the number of pairings is only N_{all} plus 1, which is not sensitive to N_{invo}. Similar reasons could also explain why the performances of NYO and YRL do not change in the same setting.

To evaluate the communication costs, we compare the size of the network name and the size of the content itself. The purpose to compare the network name is to make sure that the names generated by our scheme does not consume much more storage space than existing solutions. The size of the network name is determined by the size of the hash algorithm outputs. In PBC library[17], a data structure element_t is used to represent an element in an algebraic structure. The size of this structure is 8 bytes. Thus, for our scheme, we need 24 bytes to store the network name. Compared with this name

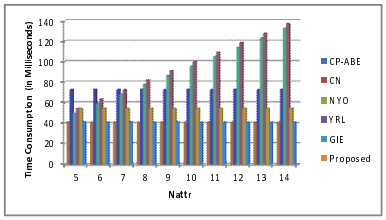

Figure 3: Nattr v.s. Time Consumption

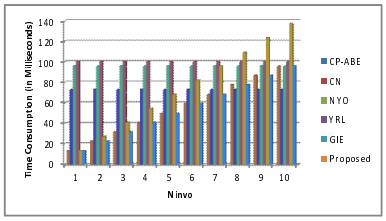

Figure 4: Ninvo v.s. Time Consumption

Table IV: Comparison of ciphertext size

Scheme	Ciphertext Size
CP-ABE	$1\mathbb{G}_1 + (2N_{ciph} + 1)\mathbb{G}_0$
CN	$1\mathbb{G}_1 + (N_{all} + 1)\mathbb{G}_0$
NYO	$\geqslant 1\mathbb{G}_1 + (2N_{all} + 1)\mathbb{G}_0$
YRL	$1\mathbb{G}_1 + (3N_{all} + 2)\mathbb{G}_0$
GIE	$N_{ciph}\mathbb{G}_1 + 3N_{ciph}\mathbb{G}_0$
Proposed	$1\mathbb{G}_1 + (2N_{ciph} + 4)\mathbb{G}_0 + N_{ciph}\mathbb{Z}_p$

size, a content in CBCB[1] is identified by a set of attributes with corresponding values. The size of this attribute set is determined by the content owners. Thus, we can model the names as a hunman-readable string of an undetermined size. NDN[3] shares a similar problem with the name size since the names in NDN also consists of a number of human-readable strings. As mentioned before, DONA[2], NetInf[4] and PURSUIT[5] share the same naming scheme. Therefore, we only use the size of DONA's name for comparison. In [2], the size of the name is confined to 40 bytes in its protocol header. Thus, the network name size in our scheme is small enough for ICN usage.

The content size of different naming schemes differs depending on the way a content is structured. However, the basic component are the same. That is a digest and the data of content. To this end, there is not much difference between the proposed scheme with existing schemes in content size.

B. Security Analysis

!!!!!

We analyze security performance of our scheme based on the attack model provided in Section III-B. In the following, we give sketches to prove that the security strength of our algorithm is no weaker than CP-ABE. Therefore, it is impossible for an attacker to retrieve the session key without satisfactory attributes. We also prove that attackers cannot gain more information from collusion attack. Furthermore, an attacker cannot confirm an attribute in decryption process if he does not own the attribute. Finally, the proposed scheme is able to guarantee forward and backward secrecy.

Theorem 1: The cryptographic strength of the proposed scheme is as good as that of CP-ABE scheme.

Proof Sketch: To prove this theorem, we need to prove that the changed components in ciphertext do not reduce the security of the proposed scheme. There are two differences between the proposed scheme and CP-ABE in ciphertext. The first one is the choice of exponents in $C_{1,n}$ and $C_{2,n}$ for each attribute A_n. In CP-ABE, the exponent, $q_y(0)$, is equal to the y-axis coordinate of a random point on a polynomial chosen for the attribute. In the proposed scheme, this value is $(I_{n-1} - I_n)t_n$ which is the difference between the current attribute secret I_n and its parent attribute secret I_{n-1} multiplied by a random value t_n. Both exponents in the proposed scheme and CP-ABE are randomized so that an attacker cannot gain any useful information from the distribution of the content names. Then assume an attacker is able to deduce the values of $(I_{n-1} - I_n)t_n$, $1 \leqslant n \leqslant m+1$, using a certain method, this attacker still cannot get the value of $(I_{n-1} - I_n)$ nor s since he has no knowledge of t_n. However, if this method works, it can also be applied to deducing the exponent in CP-ABE, which eventually leads to the leak of s using Lagrange polynomial interpolation.

The other difference is that there is an additional ciphertext $C_{3,n}$ for each attribute in the proposed scheme. If an attacker is able to retrieve the random value t_n, he is able to get the values $g^{(I_{n-1}-I_n)}$, $T_n^{(I_{n-1}-I_n)}$ and k_n. But he cannot find any useful information if he does not possess the secret information T_n and I_n associated with attribute A_n. ■

Theorem 2: The proposed scheme is secure against collusion attack.

Proof Sketch: The proposed scheme guarantees the uniqueness of intermediate decryption results for each consumer. That is in **NodeJoin** algorithm, the random value r_{UID} chosen for each entity is different and unique. If attackers combine their keys together to decrypt the same policy, the intermediate results they can get are in the form of $e(g,g)^{r_{UID}I_n}$, which are different between the attackers. Furthermore, the difficulty for an attacker (UID) to convert his intermediate result to the result of another entity (UID') equals to the difficulty to get the value $r_{UID'}/r_{UID}$ which is only known to the TTP. Thus, attackers cannot correctly recover either the intermediate results or the secret message S_k from collusion. ■

In GIE, when a decrypter successfully decrypts ciphertext corresponding to one attribute, it is able to know what the next attribute is for continuing the decryption process. Attackers can exploit such knowledge to infer or deduce more information about the targets. In the proposed scheme, this problem is solved so that the attacker cannot tell what the next attribute is if it does not own this attribute.

Theorem 3: An attacker cannot confirm attributes other than his own in decryption process.

Proof Sketch: The decryption process in the proposed scheme is conducted attribute by attribute. A decrypter is able to confirm his ownership of the next attribute if he successfully decrypts the current one. But he is unable to gain

any knowledge about the next attribute if he does not own that attribute. In fact, when an attacker successfully decrypts along the decryption path to an attribute A_n, he is able to get the value $e(g,g)^{r_{UID_{dec}}(I_{n-1})}$. He can also get $e(g,g)^{r_{UID_{dec}}}$ from $Z_{n,UID}$ and I_n. However, due to the difficulty of Discrete Logarithm Problem, the attacker is not able to deduce I_{n-1}. ■

Theorem 4: The proposed scheme guarantees forward and backward secrecy.

Proof Sketch: To maintain forward and backward secrecy for each communication group, the group session key needs to be updated by encrypting and distributing the new session key using our scheme. In addition to group communication secrecy, it is necessary to guarantee the forward and backward secrecy for each attribute key. If an entity is assigned with an attribute A_n after the network setup, it is assigned with the updated key corresponding to this attribute, i.e. $Z'_{n,UID} = (Z_{n,UID})^{I'_n/I_n}$, $k'_n = k_n + \Delta k_n$. The entity is not able to decrypt previous communications using this attribute with its current keys. This is because all the elements in its key are updated to new values except for T_n. Without the knowledge of I_n, the attacker cannot conduct any attacks as discussed in Theorem 1. This security guarantee is also applicable to forward secrecy. But for forward secrecy, the updated keys are distributed with the proposed scheme to all the nodes except for those whose attribute is revocated. ■

VI. Conclusion

!!!!!!

In this paper, we propose a novel naming scheme for ICN network. This scheme is based on a new design of ABE-based algorithm. The content names are protected based on attributes. This scheme greatly reduces the communication and computation overhead compared to existing ABE solutions. Also, this scheme is designed in a public-key pattern, making it more flexible for attribute management. From security and privacy perspective, this scheme achieves a security level as good as CP-ABE but with protection on attribute policies. It guarantees attribute anonymity with no attribute exposure. Forward and backward secrecy is achieved with a key update mechanism. Experiments and analysis confirm the effectiveness of this scheme.

References

[1] A. Carzaniga, M. Rutherford, and A. Wolf, "A routing scheme for content-based networking," in *INFOCOM 2004*, 2004, pp. 918–928.

[2] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, 2007, pp. 181–192.

[3] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, 2009, pp. 1–12.

[4] C. Dannewitz, J. Golic, B. Ohlman, and B. Ahlgren, "Secure naming for a network of information," in *INFOCOM IEEE Conference on Computer Communications Workshops , 2010*, 2010, pp. 1–6.

[5] N. Fotiou, P. Nikander, D. Trossen, and G. Polyzos, "Developing information networking further: From psirp to pursuit," in *Broadband Communications, Networks, and Systems*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer Berlin Heidelberg, 2012, pp. 1–13.

[6] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," in *Proceedings of the second edition of the ICN workshop on Information-centric networking*, 2012, pp. 55–60.

[7] N. Fotiou, G. F. Marias, and G. C. Polyzos, "Access control enforcement delegation for information-centric networking architectures," in *Proceedings of the second edition of the ICN workshop on Information-centric networking*, 2012, pp. 85–90.

[8] S. Singh, "A trust based approach for secure access control in information centric network," *International Journal of Information and Network Security (IJINS)*, vol. 1, no. 2, pp. 97–104, 2012.

[9] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the IEEE Symposium on Security and Privacy*, 2007, pp. 321–334.

[10] S. Yu, K. Ren, and W. Lou, "Attribute-based on-demand multicast group setup with membership anonymity," in *Proceedings of the 4th international conference on Security and privacy in communication netowrks*, 2008, pp. 18:1–18:6.

[11] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proceedings of the 30th Annual international conference on Theory and applications of cryptographic techniques: advances in cryptology*, 2011, pp. 568–588.

[12] S. Arianfar, T. Koponen, B. Raghavan, and S. Shenker, "On preserving privacy in content-oriented networks," in *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, 2011, pp. 19–24.

[13] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Proceedings of the 6th international conference on Applied cryptography and network security*, 2008, pp. 111–129.

[14] D. Huang, Z. Zhou, and Z. Yan, "Gradual identity exposure using attribute-based encryption," in *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, 2010, pp. 881 –888.

[15] R. Biswas, K. Chowdhury, and D. Agrawal, "Attribute allocation and retrieval scheme for large-scale sensor networks," *International Journal of Wireless Information*

Networks, vol. 13, no. 4, pp. 303–315, 2006.

[16] L. Cheung and C. Newport, "Provably secure ciphertext policy abe," in *Proceedings of the 14th ACM conference on Computer and communications security*, 2007, pp. 456–465.

[17] B. Lynn, "Pbc library the pairing-based cryptography library," in *http://crypto.stanford.edu/pbc/*, Accessed March 2013.